

# Domino & WebSphere Integration on iSeries

LP01

ITSO iSeries Technical Forum 2001

Debbie Landon  
Wilfried Blankertz



---

© 2001 IBM Corporation

## Agenda



### Overview

- Collaborative Commerce - Why Domino & WebSphere?
- Points of Integration

### Integration Topics

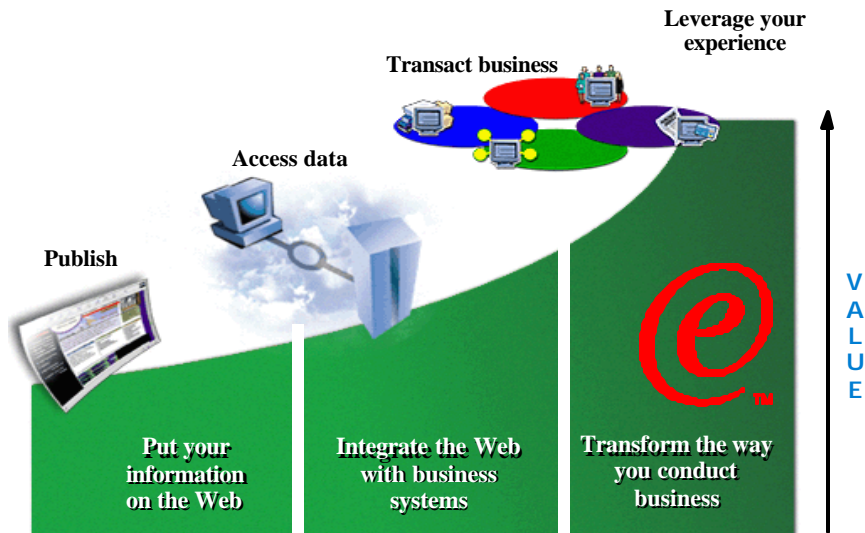
- HTTP Options
  - ▀ Domino HTTP Server Support
    - Configure WebSphere to use Domino HTTP server
  - ▀ OS/400 HTTP Server Support
    - Configure Domino to use OS/400 HTTP server
- Authentication & Directory Sharing
- Single Sign-On

### Summary / Resources

---

© 2001 IBM Corporation

## The e-business Journey



Source: McKenna Group, 1998

© 2001 IBM Corporation

## The e-business Journey



The evolution of e-business.

By now most people are familiar with the evolution of e-business. From just having a web page or home page serving static information, to integrating your web page with back-end systems and having dynamic pages updated with business information, to actually conducting business on the web. Actually changing the way business is conducted. Domino and WebSphere play an important role in helping customers to transform their business to be able to conduct their business on the web. Not only with their customers (e-commerce) but with their business partners and suppliers (business-to-business or B2B).

© 2001 IBM Corporation

## Lou Gerstner's Remarks to Analysts



"More and more of our customers realize that e-business isn't about creating one dot [com]; it's about connecting all the dots that are important in an enterprise... the Net enables and at the same time demands totally new levels of integration... Any enterprise that wants the efficiencies and cost savings of e-business has to integrate business processes and the enterprise applications that support them. "

"Middleware ... is where we have been investing for five years and investing heavily... **Middleware is what makes e-business work.** It accounts for 75 percent of what our customers spend on software... It is the critical layer to which all the new applications are being written. And we hold the number 1 or 2 share position in every major middleware category: Database, Web serving, messaging and integration, collaboration, transaction processing and systems management."

Source: Lou Gerstner security analyst briefing, 9 May 2000

© 2001 IBM Corporation

## Lou Gerstner's Remarks to Analysts



"More and more of our customers realize that e-business isn't about creating one dot [com]; it's about connecting all the dots that are important in an enterprise... the Net enables and at the same time demands totally new levels of integration... Any enterprise that wants the efficiencies and cost savings of e-business has to integrate business processes and the enterprise applications that support them. "

**Both Domino and WebSphere are an important part of IBM's e-business strategy**

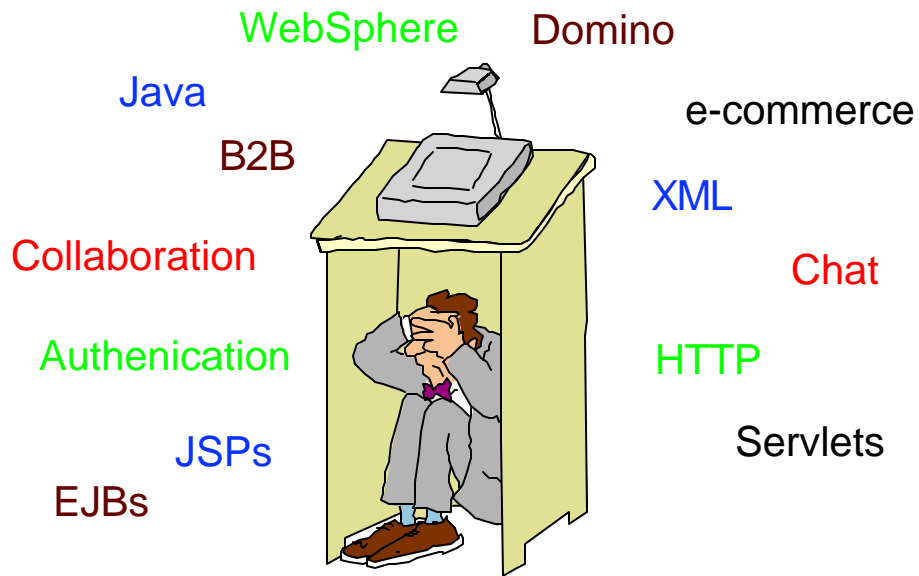
"Middleware ... is where we have been investing for five years and investing heavily... **Mi** It accounts for 75 percent of what our customers spend on software... It is the critical layer to which all the new applications are being written. And we hold the number 1 or 2 share position in every major middleware category: Database, Web serving, messaging and integration, collaboration, transaction processing and systems management."

**Domino and WebSphere are middleware -- the arena where integration happens**

Source: Lou Gerstner security analyst briefing, 9 May 2000

© 2001 IBM Corporation

Understanding infrastructure can be intimidating...



© 2001 IBM Corporation

Understanding infrastructure can be intimidating...



Understanding the infrastructure or middleware can be intimidating....lots of buzz words and acronyms. This presentation will focus on Domino and WebSphere and try to make sense of what these pieces are and where they fit.

© 2001 IBM Corporation

## Positioning Domino and WebSphere



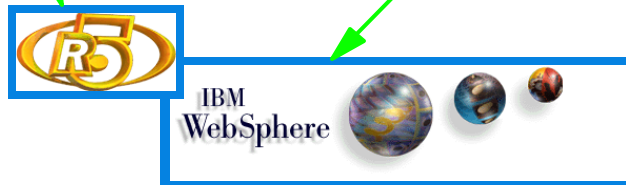
Domino - enabler for collaboration

- Business process automation
- Workflow
- Information sharing
- Communications

WebSphere Application Server

- enabler for transactions

- Business logic
- Java Programming Model (Servlet, JSP, EJB)
- Transaction processing



© 2001 IBM Corporation

## Positioning Domino and WebSphere



Web Application Servers have evolved to incorporate and support more application-specific functionality, such as business logic, workflow and collaboration. The combined Domino/WebSphere solution offers a variety of tools for choosing the right tool for the job. Companies that make the wrong choice can build an application that doesn't scale, doesn't meet required user response times, and can't adequately handle large volumes of data.

Domino is optimized for services like:

- Application services: Messaging, Directory, Security, Calendaring, Search
- Advanced services: Workflow, Content Management, Access Control, Mobile, Object Store

Domino is the premier platform for "Convergent Applications":

- Automating unstructured business processes
- Managing work and information flow (Supply Chain Management, Customer Relationship Management (CRM))
- Building Electronic Relationships through focused collaboration

Domino is more sophisticated at collaboration capabilities and is MUCH MORE than just e-mail.

WebSphere Application Server (WAS) is optimized for Infrastructure services like:

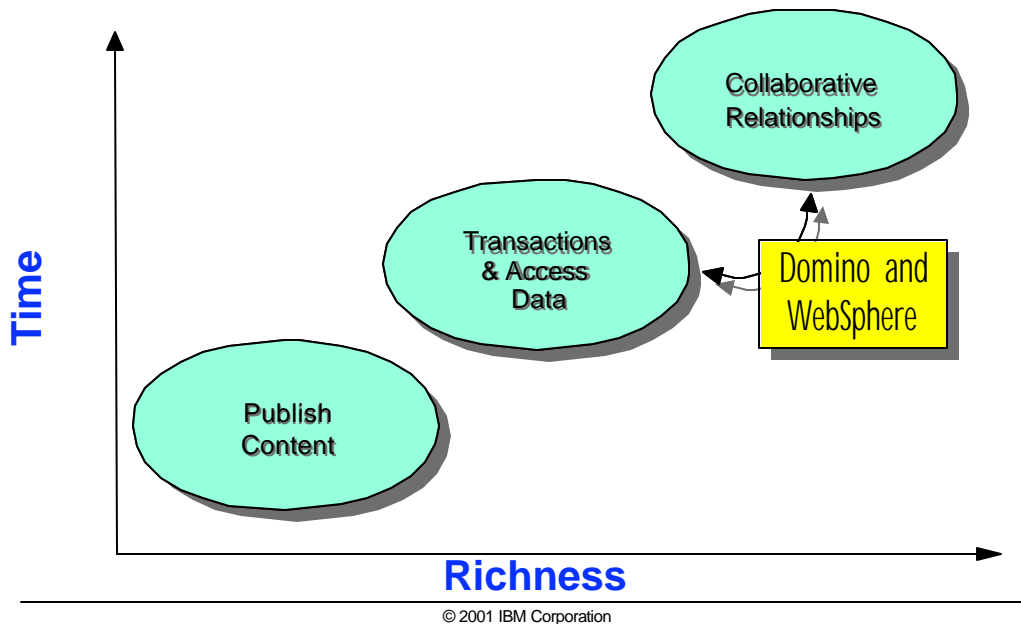
- Cross web server (IIS, Domino, iPlanet, Apache, IBM HTTP Server)
- Distributed Transaction Management, Session Management
- Java programming model (Servlet, JSP, EJB)

WebSphere is more sophisticated at transaction capabilities.

Conclusion: Domino and WebSphere have complementary strengths required for e-business today.

© 2001 IBM Corporation

## The Evolution of e-commerce



## The Evolution of e-Commerce



e-Commerce has grown up from being primarily about content to enabling transactions (the capabilities to commit purchase orders). More recently, however, commerce has been about collaborative relationships. This has come about as a result of several factors:

- First, commerce between companies or B2B Commerce has become more prevalent in the industry. B2B commerce requires much greater need for relationships and collaborative technologies to facilitate purchasing of large, complex transactions between teams of people.
- Second, sites with transactional capabilities are realizing that they could increase the amount and frequency of purchasers by assisting transactions.
- Third, as commerce sites grow up, they have realized that there is much more about the buying and selling process than the actual transaction.

e-Collaboration is a key component of e-business solutions. Domino has already proven its adaptability and flexibility to accommodate many different uses and applications within organizations and on the Web. But perhaps Domino's greatest strength in the face of today's changing business landscape is its ability to support groups of people as they invent new ways to work together. That's the secret of its leadership as a groupware environment. And now, we're seeing groupware reinvented as e-Collaboration, with Domino in the forefront as the preferred environment.

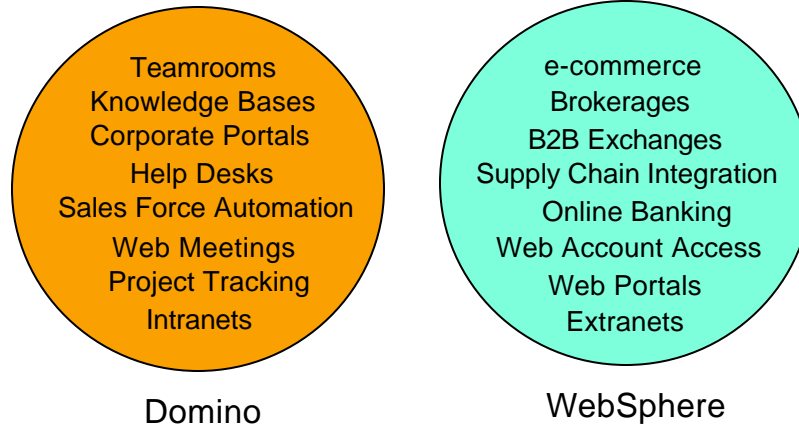
e-Collaboration includes formal applications -- notably Customer Relationship Management and Supply Chain Management. Some of the ISV solutions for CRM and SCM are built on a Domino foundation. But e-Collaboration also includes ad hoc, unstructured communities of people coming together for short projects or more permanent efforts. Think about QuickPlace or Domino discussion databases -- ideal tools for people working together.

When you build your Web site, these communities are already happening. Your knowledge experts are creating tips and techniques to put on your Web site. Visitors to your Web site are providing feedback about other information they'd like to see -- or even tips that you can post. Directly or indirectly, this is collaboration. Gradually, organizations will invent ways to make this collaboration more direct and powerful -- more beneficial to both sides. And Domino -- with its track record for versatility -- is a great tool to enable whatever form this e-Collaboration takes.

e-Collaboration applications have a couple things in common. They all focus on "linkage" between groups of people...and they all focus on the exchange of ideas and information rather than the processing of transactions and data... You begin to get the picture of where Domino fits best in the e-business world.

© 2001 IBM Corporation

## Domino & WebSphere Applications



Domino

WebSphere

© 2001 IBM Corporation

## Domino & WebSphere Applications



Both WebSphere and Domino are extremely popular tools for building applications.

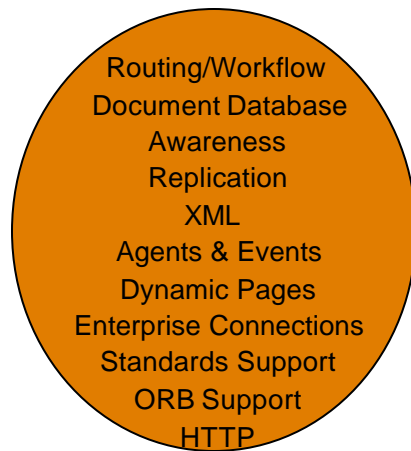
This figure shows the typical types of applications that people build with Domino or WebSphere are very different.

For Domino it's the people-centric, document-based applications such as TeamRooms, Knowledge Bases and Intranets. Domino, in a nutshell, is great when people need to deal with documents (unb-structured or semi-structured), or especially when they have to share them, or move them around in a workflow.

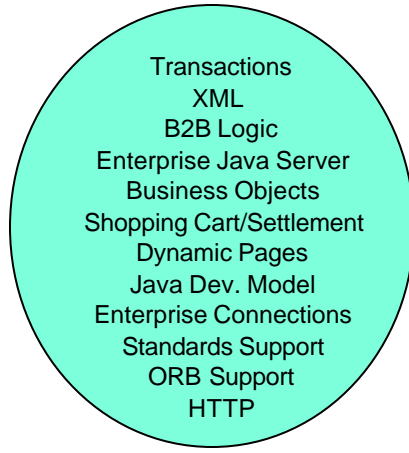
For WebSphere it's the systems-centric, data based applications such as storefronts, Supply Chain Integration and extranets. WebSphere, in a nutshell, is great when you want to extend back-end systems to the Web, as an extranet or wrapped into a useful context such as an e-commerce site or B2B Exchange.

© 2001 IBM Corporation

## Domino & WebSphere Functions



Domino



WebSphere

© 2001 IBM Corporation

## Domino & WebSphere Functions



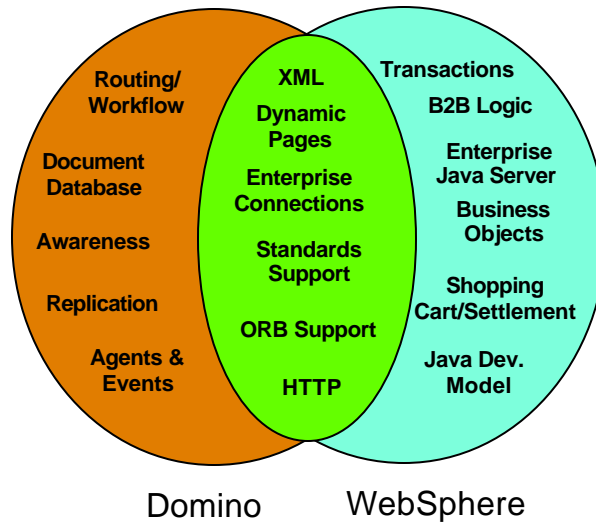
The reasons that people use Domino and WebSphere for different things is that they have different underlying capabilities.

Domino has its world-class routing and document database, its unrivaled replication and agents technology. There are Domino features around awareness and application sharing, in SameTime and QuickPlace. Then there are a whole bunch of "plumbing" things that Domino needs to have as an application server -- page rendering, enterprise integration, a development model, Internet standards support, etc.

WebSphere has rich support for transactions, a library of business objects, shopping cart services and settlement when you buy it in the WebSphere Commerce Suite configuration. And, oh yes, it too has all those other things you need when you're an application server.

© 2001 IBM Corporation

## Domino & WebSphere Integration



© 2001 IBM Corporation

## Domino & WebSphere Integration



What we've committed to doing, within the e-business framework, is to take all those common "plumbing" things and merge them, so that whether you buy Domino or WebSphere, those things will always be the same. That way, as you put together your application server infrastructure, you can know that you'll be able to easily add Domino family products to your WebSphere platform, and vice versa. The result is a very simple but very powerful dynamic. As a result of the integration, it will be easy for to, for example, add Teamrooms to your B2B Exchange. Or awareness to your e-commerce site. Or add knowledge base access to your Web account access applications. Or -- FINALLY -- integrate your intranet and extranet the way it should be done.

© 2001 IBM Corporation

## How do they work together?



### Infrastructure

- HTTP (Web Serving)
  - Domino HTTP server
  - OS/400 HTTP server
- Directory Sharing / Authentication
  - Domino or OS/400 LDAP can be used by WebSphere
- Single Sign-On
  - Shared authentication between Domino and WebSphere
    - Common cookie

### Programmatic

- WebSphere Accessing Domino
  - Notes Java Objects
    - local access
  - Notes Java CORBA objects
    - remote access
- Domino Accessing WebSphere
  - via URLs
  - Passthru HTML
  - Calling Servlets via Agents
  - Invoke EJB logic from Domino Java agent

Focus of this presentation

© 2001 IBM Corporation

## How do they work together?



As you can see there are many integration points between Domino and WebSphere. Many more than most people realize. Integration encompasses both infrastructure and programmatic topics as we see here.

The focus of this presentation today is on the Infrastructure topics, with the ultimate goal of understanding how to setup shared authentication between Domino and WebSphere. Or what is commonly called Single Sign On or SSO.

© 2001 IBM Corporation

## Objectives



### Steps to successful Domino and WebSphere Single Sign On:

1. Choose and enable HTTP server.
2. Choose LDAP server for authentication.
3. Enable security for Domino applications.
  - ─ Secure Domino application(s)
4. Enable security for WebSphere applicaitons.
  - ─ Secure WebSphere application(s)
5. Enable Single Sign On in WebSphere and Domino.

Make sure to test each step individually.

---

© 2001 IBM Corporation

## Objectives of this Presentation



The objectives of this presentation are to understand how to setup and deploy a SSO environment between Domino and WebSphere. We begin by looking at the available HTTP options for serving both Domino and WebSphere applications. We then briefly discuss enabling security for Domino and WebSphere applications with the ultimate goal of setting up a SSO between Domino and WebSphere. What SSO allows is you to sign in once and seamlessly access both Domino and WebSphere applications without being prompt for a userid and password multiple times.

---

© 2001 IBM Corporation

## Agenda



### Overview

- Collaborative Commerce - Why Domino & WebSphere?
- Points of Integration

### Integration Topics



- [HTTP Options](#)
  - [Domino HTTP Server Support](#)
    - [Configure WebSphere to use Domino HTTP server](#)
  - [OS/400 HTTP Server Support](#)
    - [Configure Domino to use OS/400 HTTP server](#)
- Authentication & Directory Sharing
- Single Sign-On

### Summary / Resources

---

© 2001 IBM Corporation

## Agenda Notes



Lets first focus on the HTTP options you have for serving both Domino and WebSphere applications. We begin by looking at using the Domino HTTP stack to serve both Domino and WebSphere applications and how you enable Domino HTTP to serve WebSphere.

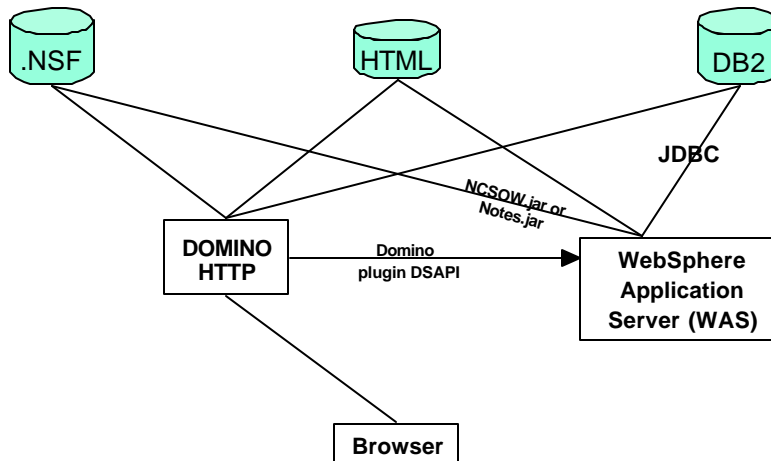
---

© 2001 IBM Corporation

## Domino HTTP Server support



- HTTP Server of Lotus Domino for AS/400 V5.0.5 or later can be used as the Web server for WebSphere Application Server V3.5.1 or later.



© 2001 IBM Corporation

## Domino HTTP Server support



Domino HTTP server has been part of the Domino Server since the availability of Lotus Notes Server R4.5, it includes these features:

- Conversion of Notes features into HTML. This conversion is governed by HTML restrictions. HTML can also be defined directly on Notes forms.
- Database ACLs are able to control access by Web browsers, as well as traditional Notes clients. Domino supports name and password authentication and Secure Sockets Layer (SSL) security.
- Support for Java applets, JavaScript, and CGI.
- Static HTML files can be published using a Domino server.
- A subset of commands are available for Domino functions, for example, opening a database, form, or view.

The IBM WebSphere Application Server is part of an architecture called the Network Computing Framework (NCF). Part of the NCF is a definition of how different Web-enabling components communicate with each other.

Therefore, the server API that connects the IBM WebSphere Application Server with an HTTP server is called the NCF Plug-in. It isolates the unique characteristics of each HTTP server in one plug-in rather than imbedding them throughout the IBM WebSphere Application Server. Thus, IBM and others can develop new plug-ins for additional HTTP servers that are independent of new releases of the IBM WebSphere Application Server. Depending on the platform, WebSphere can plug into a variety of popular Web servers, including Apache Server, IBM HTTP Server, Netscape Enterprise Server, Netscape FastTrack server, Microsoft Internet Information Server (IIS) and, Lotus Domino R5.

When the Domino HTTP stack gets a request:

- Requests for Domino objects will be routed to the Domino Server for processing.
- All other requests will be forwarded to WebSphere through the DSAPI plug-in for processing.
- WebSphere will handle requests intended for it (for example, for servlets) and return all other requests to the Domino server.

© 2001 IBM Corporation

## Domino HTTP server setup - 1



### Step 1: Update Domino Server document - Internet Protocols tab

- Specify DSAPI filter file name: `/qsys.lib/qejb.lib/domino.srvpgm`

**SERVER: DomWAS03/Dom03**

Basics | Security | Ports | Server Tasks | **Internet Protocols** | MTAs | Miscellaneous | Transactional Logging | Administration

HTTP | **Domino Web Engine** | IIOP | LDAP | NNTP

| Basics   |   | Mapping              |   |
|--|---|----------------------|---|
| Host name(s):  | <input type="text" value=""/>                                 | Home URL:            | <input type="text" value="/homepage.nsf?Open"/> |
| Bind to host name:   | <input type="text" value="Disabled"/>                         | HTML directory:      | <input type="text" value="domino\html"/>        |
| DNS lookup:  | <input type="text" value="Disabled"/>                         | Icon directory:      | <input type="text" value="domino\icons"/>       |
| Default home page:   | <input type="text" value="default.htm"/>                      | Icon URL path:       | <input type="text" value="/icons"/>             |
| Allow HTTP clients to browse databases:  | <input type="radio"/> Yes <input checked="" type="radio"/> No | CGI directory:       | <input type="text" value="domino\cgi-bin"/>     |
| Maximum requests over a single connection:   | <input type="text" value="1"/>                                | CGI URL path:        | <input type="text" value="/cgi-bin"/>           |
| Number active threads:   | <input type="text" value="40"/>                               |                      |   |
| NOTE: The following setting is no longer used in Domino. You should use it only for servers running versions prior to 4.6. |   |                      |   |
| Minimum active threads:  | <input type="text" value="20"/>                               |                      |   |
| Enable Logging To:   |   | Log File Settings    |   |
| Log files:   | <input type="text" value="Disabled"/>                         | Access log format:   | <input type="text" value="Common"/>             |
| Domlog.nsf:  | <input type="text" value="Disabled"/>                         | Time format:         | <input type="text" value="LocalTime"/>          |
|  |   | Log file duration:   | <input type="text" value="Daily"/>              |
| Log File Names   |   | Exclude From Logging |   |
|  |   |                      |   |

DSAPI = Domino Web Server API

## Domino HTTP server setup - 1



STEP 1: The filter file (domino.srvpgm) specified here is the WebSphere DSAPI plug-in for Domino for AS/400. It can pass the requests which need WebSphere Application Server to handle. This behavior is transparent to the browser user.

Originally the Domino Web Server Application Programming Interface (DSAPI) is a C API that lets you write your own extensions to the Domino Web Server. These extensions or filters let you customize the authentication of Web users. For more information about DSAPI and filters, see the Lotus C API Toolkit for Domino and Notes Release 5.0.3. The toolkit is available at: <http://www.lotus.com/techzone>. For Domino and WebSphere integration a service program is provided to allow use the Domino HTTP server for WebSphere.

## Domino HTTP server setup - 2 & 3



### Step 2: Update Domino server's NOTES.INI file and restart HTTP

- Add line to specify configuration file used by WebSphere:  
`WebSphereInit = /qibm/userdata/webasadv/default/properties/bootstrap.properties`
- Start or Restart Domino server's HTTP task:  
`load http` OR `tell http restart`

### Step 3: Update OS/400 authorities

- Grant QNOTES authority to create WebSphere log files
  - `CHGAUT OBJ('/QIBM/UserData/WebASAdv/default/logs') USER(QNOTES) DTAAUT(*RWX)`

© 2001 IBM Corporation

## Domino HTTP server setup - 2 & 3



STEP 2: There is some configuration information for WebSphere DSAPI plug-in included in the file `bootstrap.properties` so the plug-in can connect the Domino HTTP Server and WebSphere together.

Here the use is for the WebSphere Application Server's default instance. For a non-default instance, modify the line to reference the directory of the WebSphere Application Server instance. For example, if you have a instance named Nicole, you should change the line as below:

- `WebSphereInit = /qibm/userdata/webasadv/Nicole/properties/bootstrap.properties`

The DSAPI plug-in can be loaded when the Domino HTTP task is restarted.

You must then stop and start the Domino HTTP server for this change to take effect. Make sure the Domino HTTP server starts with no error messages, if there are errors you may have a typo in the `NOTES.INI` file or the change you made to the Domino server document in step1 was not correct.

STEP 3: Grant the user profile QNOTES the authority needed to be able to create the necessary WebSphere Application Server log files.

Here the use is for the WebSphere Application Server's default instance. For a non-default instance, modify the OBJ parameter to reference the directory of the WebSphere Application Server instance. For example, if you have a instance named Nicole, you should change the command as below:

- `CHGAUT OBJ('/QIBM/UserData/WebASAdv/Nicole/logs') USER(QNOTES) DTAAUT(*RWX)`

© 2001 IBM Corporation

## Agenda



### Overview

- Collaborative Commerce - Why Domino & WebSphere?
- Points of Integration

### Integration Topics

- HTTP Options
  - Domino HTTP Server support
    - Configure WebSphere to use Domino HTTP server
  - OS/400 HTTP Server support
    - Configure Domino to use OS/400 HTTP server
- Authentication & Directory Sharing
- Single Sign-On

### Summary / Resources

---

© 2001 IBM Corporation

## Agenda Notes



We will now discuss how you can create an OS/400 HTTP server instance and configure it to serve both WebSphere and Domino content.

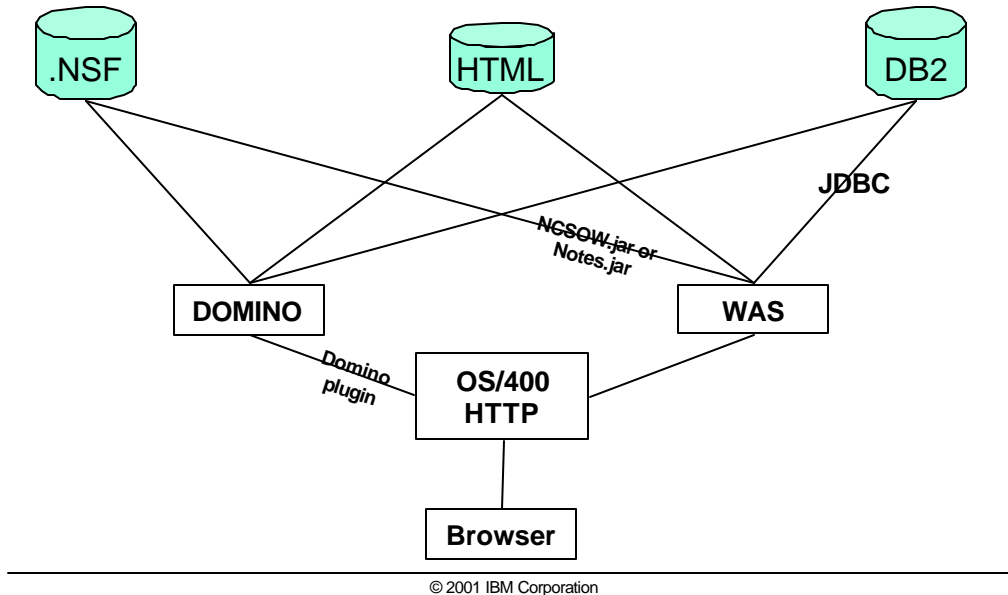
---

© 2001 IBM Corporation

## OS/400 HTTP Server support



OS/400 HTTP can be a Web Server for Domino 5.05



## OS/400 HTTP Server support



The IBM WebSphere Application Server is IBM's Java based Web application server. It is a middle tier or Web middle ware that sits in a three-tier e-business environment, between the HTTP server and the business data and logic.

The lowest tier is the HTTP server that handles requests from the browser client. The highest tier is the business database (e.g. DB2/400) and the business logic (e.g. traditional business application, such as order processing). The middle tier is IBM WebSphere Application Server, which provides a framework for consistent, architected linkage between the HTTP requests and the business data and logic.

On iSeries, the WebSphere Application Server for AS/400 was based on the IBM HTTP Server for AS/400 (V4R3 or later). The OS/400 HTTP Server needs to hand the input from the form off to the WebSphere Application Server for processing.

When the OS/400 HTTP stack gets a request:

- HTML files are returned from the file system by OS/400 HTTP Server.
- Requests to WebSphere (for example, servlets) will be routed to the WebSphere for processing and be returned by OS/400 HTTP Server.
- Requests for Domino objects will be routed to the Domino plug-in for processing and will be returned by the OS/400 HTTP stack

© 2001 IBM Corporation

## OS/400 HTTP server setup - 1



### Step 1: Configure OS/400 HTTP server instance

- Start OS/400 HTTP Administration server  
**STRTCPSVR SERVER(\*HTTP) HTTPSVR(\*ADMIN)**
- Access HTTP Administration server on port 2001 from a Web browser:  
**http://<system name>:2001**
- Click on **IBM HTTP Server for AS/400**



© 2001 IBM Corporation

## OS/400 HTTP Server setup - 1



STEP 1: In order to create an OS/400 HTTP server configuration and instance, you need to access the HTTP Administration server on the iSeries. This is a special server instance capable of allowing administrators to modify HTTP server configuration and to start, stop, and restart instances of HTTP servers remotely from the web. The HTTP Administrator server by default runs on port 2001. Point your Web browser to **http://<system name>:2001** where <system name> is the host name of your iSeries system.

You are prompted for an iSeries user ID and password. This user ID must have \*ALLOBJ authority.

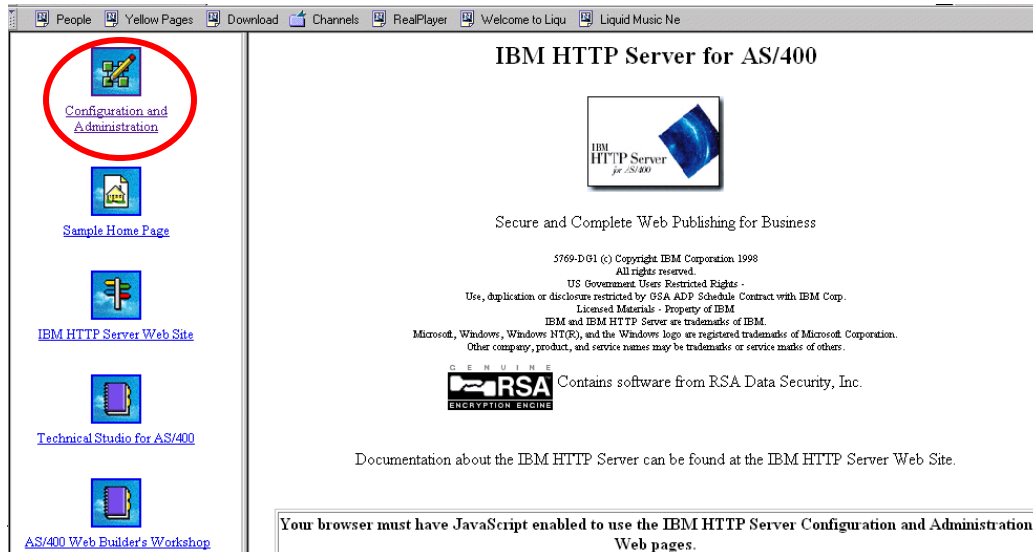
© 2001 IBM Corporation

## OS/400 HTTP Server setup - 2



### Step 2: IBM HTTP Server for AS/400 web page

- Click on [Configuration and Administration](#)



## OS/400 HTTP Server setup - 2



STEP 2: Click on **Configuration and Administration** link to continue.

## OS/400 HTTP Server setup - 3



### Step 3: Create HTTP Configuration

- Click on [Configurations](#) and then click on [Create Configuration](#)
- Enter a name for your HTTP Configuration, click [Apply](#)

Configuration:

☒ Create empty configuration

☐ Create based on existing configuration:

© 2001 IBM Corporation

## OS/400 HTTP Server setup - 3



STEP 3: You now need to create a HTTP configuration. On the IBM HTTP Server Configuration and Administration web page that is displayed, from the navigation frame on the left side of the window, click on **Configurations**, then click on **Create Configuration**.

The right frame will now prompt you for a configuration name. For purposes of this presentation, we leave the **Create Empty Config** radio button selected and click **Apply**.

**NOTE:** You must click on **Apply** whenever you make changes in this environment or the changes will not be saved!

Once the configuration is successfully created, you should see a message on the page indicating the HTTP server configuration was successfully created.

© 2001 IBM Corporation

## OS/400 HTTP Server setup - 4



### Step 4: Update HTTP Configuration - Basic information

- Select newly created HTTP configuration and click on [Basic](#)
- Enter iSeries host name and default port, click [Apply](#)

IBM HTTP Server for OS/400

Configuration and Administration

Basic

Configuration: DOMWASXX

Host name: as22

Bind options: ☐ Bind server to host IP address  
☒ Bind server to all local IP addresses

Default port: 80xx

User: %SERVER%

© 2001 IBM Corporation

## OS/400 HTTP Server setup - 4



STEP 4: Once you have successfully created a HTTP configuration you need to update some information for this configuration. Select your HTTP configuration from the drop-down box in left navigation frame and click on **Basic**.

On the Configuration and Administration Basic web page that is displayed, enter the name of your iSeries host name (or iSeries system name) and the default port this HTTP configuration will listen on. Notice that you can bind this HTTP configuration to a specific TCP/IP address or have this configuration listen on all available interfaces. Once you are finished updating this information, click on **Apply**.

**NOTE:** Remember, you must click on **Apply** whenever you make changes in this environment or the changes will not be saved!

Once the configuration is successfully updated, you should see a message on the page indicating the HTTP server configuration file was successfully updated and that any HTTP server instances using this configuration must be stopped and started for the changes to take affect.

## OS/400 HTTP Server setup - 5



### Step 5: Update HTTP Configuration - Methods information

- Click on [Request Processing](#), then [Methods](#)
- Select methods of GET and POST. click [Apply](#)

Configuration and Administration

Methods

Configuration: DOMWASXX

| Enable methods                              | Optional method handler |
|---|-------------------------|
| <input type="checkbox"/> CONNECT            |                         |
| <input type="checkbox"/> DELETE             |                         |
| <input checked="" type="checkbox"/> GET     |                         |
| <input checked="" type="checkbox"/> HEAD    |                         |
| <input checked="" type="checkbox"/> OPTIONS |                         |
| <input checked="" type="checkbox"/> POST    |                         |
| <input type="checkbox"/> PUT                |                         |
| <input checked="" type="checkbox"/> TRACE   |                         |

User-defined method handlers:

© 2001 IBM Corporation

## OS/400 HTTP Server setup - 5



STEP 5: Now you need to enable the HTTP methods required to allow the HTTP server to process CGI and then pass it to WebSphere. From the left navigation frame, click on **Request Processing** and then **Methods**.

On the Configuration and Administration Methods web page that is displayed, select the methods of GET and POST. Leave the other methods as is and click on **Apply**.

**NOTE:** Remember, you must click on **Apply** whenever you make changes in this environment or the changes will not be saved!

Once the configuration is successfully updated, you should see a message on the page indicating the HTTP server configuration file was successfully updated and that any HTTP server instances using this configuration must be stopped and started for the changes to take affect..

© 2001 IBM Corporation

## OS/400 HTTP Server setup - 6



### Step 6: Update HTTP Configuration - Java servlets information

- Click on [Java Servlets](#)
- Click on [WebSphere version 3](#) and select WebSphere domain, click [Apply](#)

Global server parameters

Server Instances

Configurations:

DOMWASXX

Basic

CGI

Create configuration

Delete configuration

Directories and Welcome Page

Display configuration

Error message customization

**Java servlets**

Languages and Encoding

LDAP

Logging

Log Reporting

Meta-information

PICS Local

PICS Third-Party

Protection

Proxy Settings

Configuration and Administration

Java servlets

Configuration: DOMWASXX

☐ Disable servlets and JavaServer pages (JSP)

☐ WebSphere version 1 or version 2

☐ Servlet URL invocation

☐ JavaServer pages (JSP)

☒ WebSphere version 3

☒ Servlets and JavaServer pages (JSP)

WebSphere domain: WAS01

Apply Reset

© 2001 IBM Corporation

## OS/400 HTTP Server setup - 6



STEP 6: Next you need to add support for Java servlets and Java Server Pages (JSPs). From the left navigation frame, click on **Java Servlets**. On the Configuration and Administration Java Servlets web page that is displayed, click on the **WebSphere version 3** radio button and select your WebSphere domain configuration from the drop-down box. Click on **Apply**. This will add the WebSphere version 3 routing entries to your HTTP configuration file.

**NOTE:** Remember, you must click on **Apply** whenever you make changes in this environment or the changes will not be saved!

Once the configuration is successfully updated, you should see a message on the page indicating the HTTP server configuration file was successfully updated and that any HTTP server instances using this configuration must be stopped and started for the changes to take affect. The message will also tell you to go to the Application Server Manager to do additional configuration.

© 2001 IBM Corporation

## OS/400 HTTP Server setup - 7



### Step 7: Update HTTP Configuration - Request Routing information

- Click on [Request Processing](#) then [Request Routing](#)
- Add 1 entry for serving HTML files and applets

| Action | URL Template | Replacement file path                         |
|--------|--------------|---|
| Pass   | /html/*      | /QIBM/UserData/WebASAdv/ <u>WASxx</u> /html/* |

© 2001 IBM Corporation

## OS/400 HTTP Server support setup 7



STEP 7: Next you need to add routing entries for serving HTML files and applets and for serving Domino files. From the left navigation frame, click on **Request Processing** and then click on **Request Routing**.

On the Configuration and Administration Request Routing web page that is displayed, add the following entry for serving HTML files and applets:

| Action | URL Template | Replacement file path                         |
|--------|--------------|---|
| Pass   | /html/*      | /QIBM/UserData/WebASAdv/ <u>WASxx</u> /html/* |

WASxx is your WebSphere Application Server instance.

**NOTE:** You must click on **Apply** to add each individual entry.

Once the configuration is successfully updated, you should see a message on the page indicating the HTTP server configuration file was successfully updated and that any HTTP server instances using this configuration must be stopped and started for the changes to take affect.

© 2001 IBM Corporation

## OS/400 HTTP Server setup - 8



### Step 8: Update HTTP Configuration - Request Routing information

- Click on [Request Processing](#) then [Request Routing](#)
- Add 3 entries for serving Domino files:

| Action  | URL Template | Replacement file path                        | CGI Conversion Mode (in/out) |
|---------|--------------|--|------------------------------|
| Service | *.nsf*       | /QSYS.LIB/QNOTES.LIB/LIBHTTPX.SRVPGM:Service | %%BINARY/MIXED%%             |
| Pass    | /icons/*     | /<Domino server data directory>/domino/icons |                              |
| Pass    | /domjava/*   | /<Domino server data directory>/domino/JAVA  |                              |

© 2001 IBM Corporation

## OS/400 HTTP Server support setup 8



STEP 8: Next you need to add routing entries for serving Domino files. From the left navigation frame, click on **Request Processing** and then click on **Request Routing**. Now add the following three entries for serving Domino files:

| Action  | URL Template | Replacement file path                        | CGI Conversion Mode (in/out) |
|---------|--------------|--|------------------------------|
| Service | *.nsf*       | /QSYS.LIB/QNOTES.LIB/LIBHTTPX.SRVPGM:Service | %%BINARY/MIXED%%             |
| Pass    | /icons/*     | /<Domino server data directory>/domino/icons |                              |
| Pass    | /domjava/*   | /<Domino server data directory>/domino/JAVA  |                              |

<Domino server data directory> is the data directory of your Domino server and the settings here assumes your Domino server uses the default directories for its icons and java applets.

**NOTE:** You must click on **Apply** to add each individual entry.

Once the configuration is successfully updated, you should see a message on the page indicating the HTTP server configuration file was successfully updated and that any HTTP server instances using this configuration must be stopped and started for the changes to take affect.

© 2001 IBM Corporation

## OS/400 HTTP Server setup - 9



### Step 9: Update HTTP Configuration - Server API Application Processing information

- Click on [Server API Application Processing](#)
- Add 2 entries to install the Domino plugin for the OS/400 HTTP server:

| Step       | Application path and file name                  |
|------------|---|
| ServerInit | /QSYS.LIB/QNOTES.LIB/LIBHTTPX.SRVPGM:ServerInit |
| ServerTerm | /QSYS.LIB/QNOTES.LIB/LIBHTTPX.SRVPGM:ServerTerm |

© 2001 IBM Corporation

## OS/400 HTTP Server setup - 9



STEP 9: Next you need to specify the Domino Plug-in for the OS/400 HTTP server. From the left navigation frame, click on **Server API Application Processing**.

On the Configuration and Administration Server API Application Processing web page that is displayed, add the following two entries for installing the Domino plug-in:

| Step       | Application path and file name                  |
|------------|---|
| ServerInit | /QSYS.LIB/QNOTES.LIB/LIBHTTPX.SRVPGM:ServerInit |
| ServerTerm | /QSYS.LIB/QNOTES.LIB/LIBHTTPX.SRVPGM:ServerTerm |

**NOTE:** Remember, you must click on **Apply** whenever you make changes in this environment or the changes will not be saved!

Once the configuration is successfully updated, you should see a message on the page indicating the HTTP server configuration file was successfully updated and that any HTTP server instances using this configuration must be stopped and started for the changes to take affect.

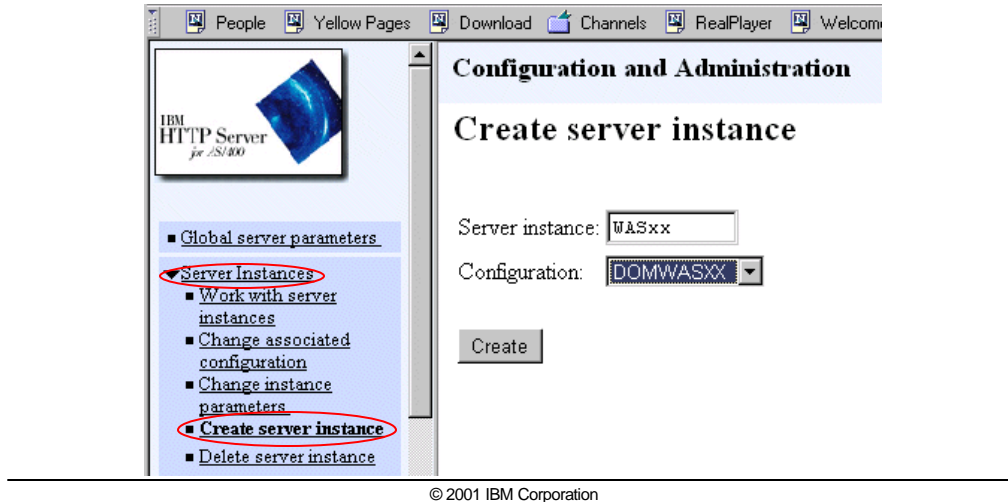
© 2001 IBM Corporation

## OS/400 HTTP Server setup - 10



### Step 10: Create an HTTP server instance

- Click on [Server Instances](#) and then [Create server instance](#)
- Name the HTTP server instance and select the HTTP configuration file



© 2001 IBM Corporation

## OS/400 HTTP Server setup - 10



STEP 10: Next you need to either create an HTTP server instance or stop and start an existing HTTP server instance. For this presentation, our example creates a new HTTP server instance to use the newly created HTTP configuration file. From the left navigation frame, click on **Server Instances** and then click on **Create Server Instance**.

On the Configuration and Administration Create Server web page that is displayed, name your HTTP server instance and select your HTTP configuration file. Click on **Create**.

© 2001 IBM Corporation

## OS/400 HTTP Server setup - 11 & 12



### Step 11: Reconfigure your Domino server

- ENDDOMSVR SERVER(<Domino server>)
- CHGDOMSVR  
SERVER(<Domino server>) WEB(<OS/400 HTTP instance>)
- STRDOMSVR SERVER(<Domino server name>)

### Step 12: Start your HTTP server

- STRTCPSVR SERVER(\*HTTP) HTTPSVR(<OS/400 HTTP instance>)

### Recommendations:

- Always start Domino **before** starting OS/400 HTTP server.
- Similarly, always end OS/400 HTTP server before ending Domino.

---

© 2001 IBM Corporation

## OS/400 HTTP Server setup - 11 & 12



STEP 11: Replace the tags with the values you actually use. You can change Domino Server configuration even when Domino Server is still running. After restarting the Domino server it now uses the OS/400 HTTP server.

STEP 12: Start the OS/400 HTTP Server instance from either the Web Administration or from the OS/400 command of Start TCP Server (STRTCPSVR). You can monitor the HTTP server jobs using the Work with Active Job (WRKACTJOB) command of: WRKACTJOB JOB job(<AS/400 HTTP server name>)

NOTE: It is recommended that you always start the Domino server BEFORE starting the OS/400 HTTP server. Similarly, you should always end the OS/400 server before ending the Domino server.

---

© 2001 IBM Corporation

## Agenda



### Overview

- Collaborative Commerce - Why Domino & WebSphere?
- Points of Integration

### Integration Topics

- HTTP Options
  - ─ Domino HTTP Server Support
    - Configure WebSphere to use Domino HTTP server
  - ─ OS/400 HTTP Server Support
    - Configure Domino to use OS/400 HTTP server



- [Authentication & Directory Sharing](#)
- Single Sign-On

### Summary / Resources

---

© 2001 IBM Corporation

## Agenda



In this section of the presentation will we BRIEFLY discuss security, specifically the subject of authentication, and then cover directory sharing specifically the subject of LDAP. Our goal for discussing these topics is a basis for the discussion of Single Sign-On between Domino and WebSphere in the following section.

---

© 2001 IBM Corporation

## Web Authentication



### Authentication verifies user identity

- Who are you?
- How can you prove it?

### Authentication involves verifying credentials.....

- Name & password, SSL certificate
- Cookie, token, ticket
- Fingerprint, voiceprint

### ...against an authority...

- Directory
- SSL Keyring
- Operating system
- Custom application



### ...resulting in an "authenticated credential".

© 2001 IBM Corporation

## Web Authentication



Authentication is the process used by a system to identify a user. It is intended to be a way for the user to prove to the system who the user is, in a way that the system can be confident that the user is really who they say they are. It does not imply any privileges that allow any action to be performed, but being authenticated means that the system knows who you are, and trusts that you are who you say you are. Generally, authentication means that the user presents some form of identification (called credentials) to the system, and the system validates those credentials against some authority, such as a directory or SSL keyring. What credentials are required and what authority is used are dependent on the system, and how it has been set up.

The credentials the user presents to the system may take several forms, and many systems allow customization that permits one or more of these to be used. The first is something you know - like a userid and password. The user presents this information to the system, and the system looks it up in a registry to be sure that what you told it matches its records. This is generally called "Basic Authentication". The downside is that it can be vulnerable to spying. Also, people forget their password, especially if they have a lot of them, or they write them down and hide them in places where they can be easily found, or they choose easy to remember passwords that can be guessed.

Rather than using something in the user's head, some systems require something physical as a credential - a digital certificate, dongle, or some kind of token, often in combination with a password. For those of you familiar with Domino, the Notes client uses a system like this - you have a Notes ID which is "unlocked" using a password. This may be less vulnerable to sharing, but can be lost. It also in many cases less convenient to the user, and more difficult to administer, and there's usually a hardware cost involved.

Sounds like Star Trek stuff, but using biometrics for authentication is becoming reality for some highly security conscious organizations, and products are starting to become more generally available. Using a physical characteristic such as a fingerprint or retina scan is much harder to forge, assuming the technology is good enough to prevent outright mistakes, but there are some privacy concerns here - do I really want to share my fingerprint with every system I want to talk to? And if someone needs my thumb to access the system, how desperate are they to get it?

You may choose to implement Multiple-Factor Authentication, which is a way to add security by combining the methods above - but be careful it is not subverted. Writing my PIN number on my ATM card gets me right back to single factor!

Once credentials have been validated by the system, they are considered "authenticated credentials", and are then used as identification to the system to determine the ability to perform actions. You should be conscious of how these authenticated credentials flow across a network - if someone can pick them off, they can be used to impersonate one user by another. That is why you may want to use encryption over the network, and many systems allow the process of authentication to initiate an encrypted session.

© 2001 IBM Corporation

## Web Authentication Challenge Types



### No authentication -- Anonymous

- "I don't care who you are"
- May want to protect certain resources against anonymous access

### Basic authentication -- user name & password

- "Tell me a secret only you would know"

### Basic authentication with encrypted channel

- "Tell me a secret in our secret code"

### Client certificate authentication

- "May I please see your badge?"

---

© 2001 IBM Corporation

## Web Authentication Challenge Types



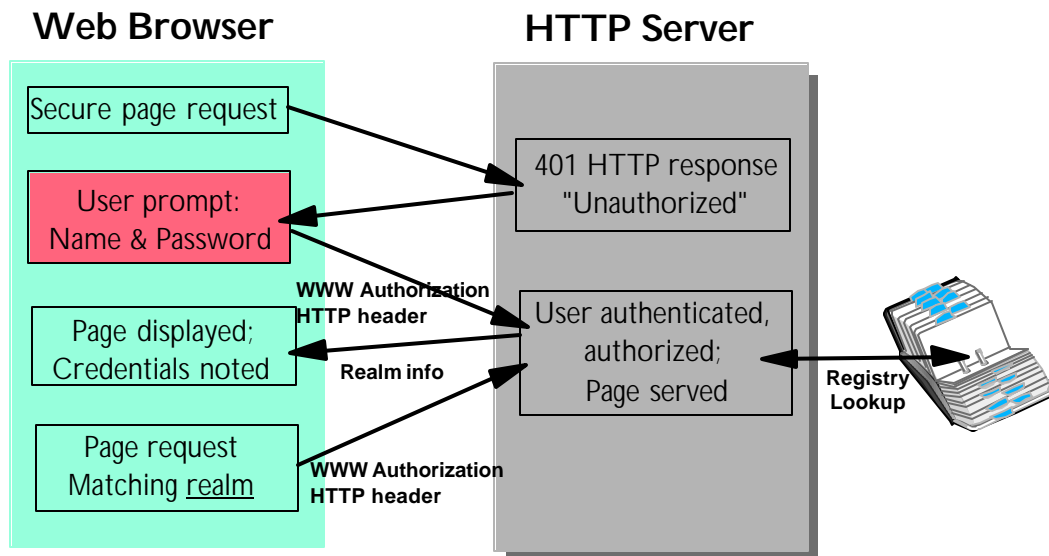
Some types of challenges, how the system asks who you are:

- No authentication - None - may not need to know, but if a resource is protected, and the system never asks who you are, you will never be allowed to see that resource. Without a challenge, you are considered anonymous.
- Basic - user name and password are required. Not extremely secure. May be done by facilities in the HTTP protocol or custom.
- Basic with SSL - the communication over the wire may be encrypted. This may be forced at the time authentication happens.
- Certificate - an X.509 (or other type) certificate is presented. If the system trusts whoever created the certificate, you are considered OK. Very secure if properly configured.

---

© 2001 IBM Corporation

## How Basic Web Authentication Works



© 2001 IBM Corporation

## How Basic Web Authentication Works



The following steps are how a basic Web authentication works:

- Web Browser requests a secure page.
- HTTP server returns a 401 HTTP "unauthorized" response.
- 401 causes Web browser to produce name/password dialog.
- User keys name/password and Web browser sends it to the HTTP server.
- HTTP server validates against a registry and returns the requested page.
- Web Browser notes success for realm.
- When user requests another page in that realm, Web browser sends name and password again in header, user is not prompted again.

A realm is a web concept that defines the "space" for which an authentication is good. Once you have authenticated for a realm, you are considered authenticated for everything within that realm. When you go to another realm, you may have to authenticate again. In HTTP, by default, the browser remembers what credentials worked in what realm, and every time it talks to a server in that realm, it sends those credentials again in the header - not under the user's control. The only control is setting up what servers are in what realm.

In Domino, you cannot set a realm to be larger than a single server. Before Domino R5, the realm in Domino consisted of the subdirectory and everything below it. This means that if you first authenticated for a resource below the root, and then went up a level, you were asked to authenticate again. In R5, you can define the realm to be one or more subdirectories, but prior to Domino R5.05 you were still limited to a single server. This can cause problems when doing multiple servers for backup or load balancing and may require reauthentication.

© 2001 IBM Corporation

## Domino Authentication Challenge Types



### Basic authentication

- Credentials sent by Web browser with every request
- Simple 64-bit encoding
- May require encrypted channel (server side SSL)

### Session based authentication

- Cookie based
- Single or Multi-server

### Client certificate authentication

- Client side SSL

### Custom

- DSAPI



© 2001 IBM Corporation

## Domino Authentication Challenge Types



Domino allows for the challenge types discussed above...plus, new with R5, session authentication allows for a custom login page, credentials stored in a cookie, and session tracking & expiration. Until 5.05, only a single server was supported. Slightly more secure than basic authentication since the user name and password are encrypted and not transmitted in clear text.

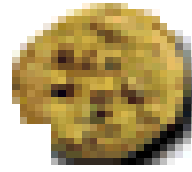
Also custom - the Domino server API (DSAPI) is an exit point that allows you to create your own authentication logic.

© 2001 IBM Corporation

## HTTP Cookies



- Commonly used today in customized solutions.
- Web browser automatically returns cookie to specified server or domain.
- Each Web server or application visited checks cookie through APIs.
- Often uses central authentication resource/server.



---

© 2001 IBM Corporation

## HTTP Cookies



Commonly used today in customized solutions.  
Web browser automatically returns a cookie to specified server or domain.  
Each Web server or application visited checks the cookie through its APIs.  
Often uses central authentication resource/server.

---

© 2001 IBM Corporation

## Domino R5 HTTP Session Authentication



**Cookie created at authentication**

**Customized login form**

**Name and password only passed once**

- Credential sent every time

**User sessions are displayed in Domino server console**

- tell http show users

**Domino R5.05 introduces multi-server option**

---

© 2001 IBM Corporation

## Domino R5 HTTP Session Authentication



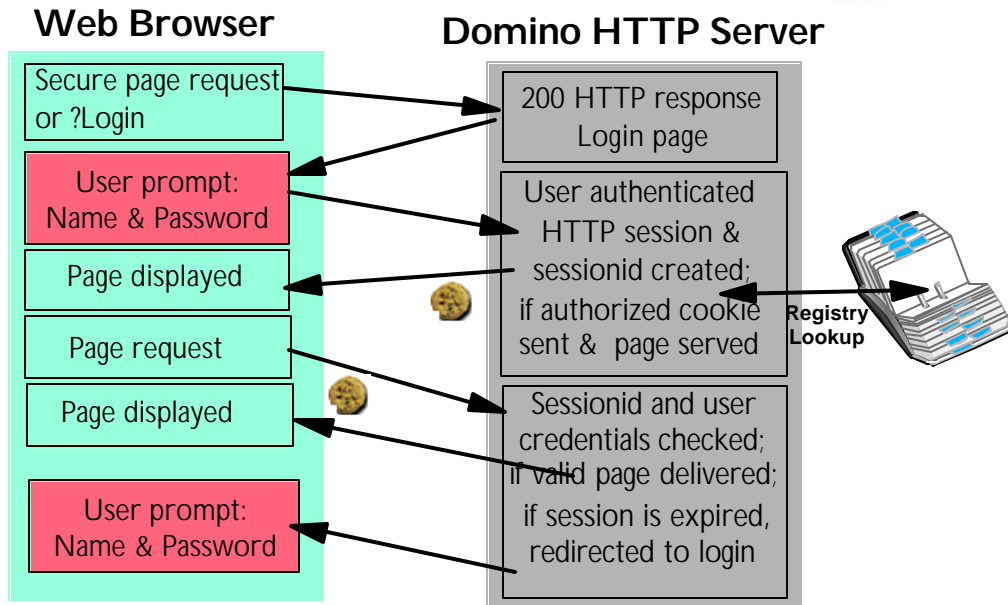
New with Domino R5, when a user authenticates, a unique session id is created on the Domino server and information is passed to user in cookie. When the user returns, the Domino server checks validity of the session id. User logout causes the session to be invalidated. The cookie is destroyed when browser closes and is only valid for the issuing server. Single server session authentication (formerly "enabled" in Domino server document) session id is kept on the server and not known by any other server. Also a Domino server reboot loses sessions.

As of Domino R5.05, multi-server support is provided - cookie has changed to format that allows authentication to work across Domino servers, as well as with WebSphere.

---

© 2001 IBM Corporation

## How Domino HTTP Authentication Works



© 2001 IBM Corporation

## How Domino HTTP Authentication Works

Authentication is triggered by a user accessing a protected resource or logging in on a URL. Domino sends a login page (may be customized in Domcfg.nsf). User enters name and password. Server authenticates user and creates a cookie (DomAuthSessid for single server, LTPAToken for multi-server). User is redirected to page requested. When user requests another page, server checks cookie for validity (expired, or max sessions reached), returns resource or requests for new logon.

© 2001 IBM Corporation

## WebSphere Authentication Settings



### Challenge Types (with or without SSL)

- None
- Basic (User Name & Password)
- Certificate
- Custom

### Authentication Mechanism

- Local operating system
- Lightweight Third-Party Authentication (LTPA)
  - Token expiration
  - Single Sign-On (SSO)

---

© 2001 IBM Corporation

## WebSphere Authentication Settings



Basically WebSphere allows the same authentication challenge types as Domino.

Security in WebSphere must be turned on globally - it's off by default. Because turning it on protects the WebSphere Administrative console, you have the potential of locking your self out, so take a backup before you turn it on - can use this to rebuild your system if needed. Once you have turned global security on, you can override some defaults like realm and challenge type by application.

WebSphere allows authentication using local operating system criteria, or Lightweight Third-Party Authentication or LTPA, which uses an LDAP(Lightweight Directory Access Protocol) directory. With LTPA, you can configure how long the token is good for, and whether SSO is to be used. You also configure which registry is used for validation (O/S or which LDAP for LTPA).

---

© 2001 IBM Corporation

## Lightweight Third-Party Authentication



### What is Lightweight Third-Party Authentication (LTPA)?

- Framework to achieve authentication and delegation within an enterprise
- Proposed standard
- LTPA server does validation using LDAP user registry
- Successful authentication results in creation of LTPA token
- Allows authentication once per session with multiple servers

---

© 2001 IBM Corporation

## Lightweight Third-Party Authentication



It's becoming more common in the industry to allow a third party to provide the means by which authentication is done so everyone doesn't have to roll their own. The third party provides a "stamp of approval" (kind of like a user listing), the assumption being that the third party did the research, so you don't have to.

LTPA was an attempt at another standard, but it didn't really catch on, so it's still proprietary - common only to WebSphere, now Domino, and a few other products, but being phased out in the next release of WebSphere (Will still work, but not being enhanced - direction is to head toward DCE/Kerberos/J2EE standards)

Using LTPA authentication mechanism, successful authentication causes creation of a token that contains name and password info. This token can be passed to any server that can understand it to prove that the user has been authenticated.

This is used with Single Sign-On.

---

© 2001 IBM Corporation

## Registry



**Validates credentials presented at authentication challenge.**

**For basic challenge, must contain name and password.**

**Name translation may occur.**

- For example sign on with shortname, name for authorized is full name.

|            |  |
|------------|--|
| User name: | Charles Cunningham/Rochester/IBM<br>Charles Cunningham |
| Shortname: | ccunning   |

---

© 2001 IBM Corporation

## Registry



A registry is used for validation. The name and password presented are compared to a directory. Name translation may occur - e.g. You may sign on as ccunning, the Directory may translate to Charles Cunningham/Rochester/IBM as the authenticated credential. Authenticated credential is what is presented for authorization and what needs to be in ACL/authorized users list.

Just a reminder that a directory may contain more than one version of a user's name, user may present any of them, but usually a specific one becomes authenticated credential (for the Domino Directory, it is the first value in User Name field).

---

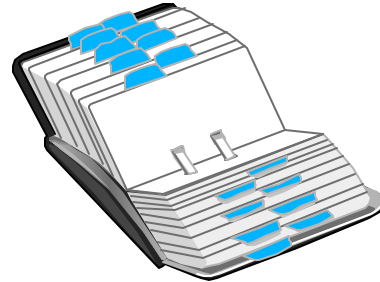
© 2001 IBM Corporation

## Directories for Web Authentication



### Directories used by Domino

- Domino Directory
  - Users with internet passwords
- Third Party LDAP
  - Using Directory Assistance
- Other (custom)
  - Using Domino server DSAPI API



### Directories used by WebSphere

- Operating System
  - OS/400 Profiles
- LDAP (using LTPA)
  - IBM SecureWay Directory
  - Lotus Domino
  - Netscape Directory Server
  - Microsoft Active Directory
- Other (custom)

© 2001 IBM Corporation

## Directories for Web Authentication



### Directories used by Domino for Web Authentication

Domino allows use of its own directory, or an LDAP compliant directory via directory assistance for web authentication (or anything you want if you code it in DSAPI)

### Directories used by WebSphere for Authentication

Operating system, or LDAP compliant - specific list supported, but you can customize for others.

© 2001 IBM Corporation

## Directories for Web Authentication



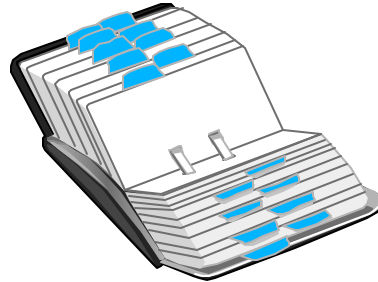
### Directories used by Domino

- Domino Directory
  - Users with internet passwords
- Third Party LDAP
  - Using Directory Assistance
- Other (custom)
  - Using Domino server DSAPI API

### Directories used by WebSphere

- Operating System
  - OS/400 Profiles
- LDAP (using LTPA)
  - IBM SecureWay Directory
  - Lotus Domino
  - Netscape Directory Server
  - Microsoft Active Directory
- Other (custom)

Single Sign-On



© 2001 IBM Corporation

## Directories for Web Authentication



For Single Sign-On, Domino and WebSphere need to use an LDAP server.

© 2001 IBM Corporation

## What is LDAP?



**LDAP = Lightweight Directory Access Protocol**

**Protocol for accessing directories**

- It is NOT a directory program!

**Designed for ease of use and performance**

- Started as a front-end to X.500 Directory Access Protocol (DAP).\
- Designed to access 90% of a X.500 directory functionality
  - At 10% of the overhead cost
- Unhooked from X.500 directory for accessing multiple directories

**Defines standard way to search and manage directory entries**

- Users & groups
- Devices
- Applications

---

© 2001 IBM Corporation

## What is LDAP?



LDAP (Lightweight Directory Access Protocol) is standard protocol for accessing directories. It is important to remember that LDAP is not a directory or server. In another words, if you have LDAP capabilities, it does not mean that you have a directory. But if you have an LDAP directory, you have a directory that can be accessed via specified interfaces - better known as LDAP application program interfaces (API). The location where the LDAP services are running is often referred to as the LDAP server. For example, if Domino server XYZ is running LDAP services it will would be referred to as the LDAP server even though it is actually a Domino server running multiple services - one of which is LDAP.

LDAP was originally developed to be a front-end to the X.500 Directory Access Protocol (DAP) because DAP's front-end was bigger than what would fit on most user's workstations. To reduce the amount of code required to run on the user's workstation they could have either taken away functions from the X500 directory or develop a directory front-end to fit on user's workstations. There choice was to design a directory front-end. The front-end design point was to access 90 percent of the X.500 Directory functions with only 10 percent of the overhead cost. They were actually successful at doing this via the LDAP code.

It was successful enough that it was decided that they should not require the overhead and complexities of X.500 protocol. That is when they made the lightweight front-end into its own protocol for accessing multiple kinds of directories instead of a front-end to the X.500 Directory Access Protocol. HOWEVER, it must be kept in mind that it still functions as a FRONT-END to directories and does not REPLACE directories.

The protocol defines a standard way to search and manage entries in a directory. Entries in a directory could be users, groups, devices, or application data

---

© 2001 IBM Corporation

## Why use LDAP?



### Open Internet standard

- Developed by Internet Engineering Task Force (IETF)
- TCP/IP, DNS/SMTP, NNTP, SNMP, HTTP, etc...
- Vendor neutral standards group

### Ability to access & update directory data

- Active and current information
- Not a transaction-based database

### Rich API interface

- Maps to LDAP supported operations
- Authentication mechanism - bind

**Required for Single Sign-On between Domino & WebSphere!**

---

© 2001 IBM Corporation

## Why use LDAP?



There are other directory protocols on the market but one of the key reasons why LDAP is emerging as the internet standard is because it is an open protocol standard developed by the Internet Engineering Task Force (IETF). And they do have some influence because this is not the first protocol the IETF has established standards for. For example, they are the ones that established the standards for protocols such as TCP/IP, DNS, SMTP, NNTP, SNMP, and HTTP. And they are considered to be a vendor neutral group. So it is a well established and respected group.

Additionally, companies like Netscape and Microsoft Internet Explorer are already LDAP-enabled. LDAP standards provide for directory information to be queried and updated. This allows the directory to be active and flexible enough to change as the environment changes. For example, in a banking environment a user's information could be updated if they qualified for a loan or opened an account.

However, it should always be kept in mind that an LDAP directory is not a data base file that should be updated constantly or used as a file system. It is not a transaction-based service. Even though there are standards in the protocol to allow you to update a directory data base and store large objects in the directory, directories should be looked at more as read and search data base on user information that does not change constantly (i.e., phone numbers, e-mail addresses).

Nor should it be considered to be a 'Do Everything' protocol. It should still be looked at as a front-end to a directory. The protocol standards are expanding to include more functions but it is not a replacement to X.500 Directory protocol or other full-function directories.

LDAP standards provide for a rich set of capabilities. One of these capabilities, BIND, allows LDAP services to be used to authenticate internet users.

---

© 2001 IBM Corporation

## Domino LDAP



### LDAP is part of the Domino server

- No separate purchase or install required
- LDAP V2 (Domino 4.6) & LDAP V3 (Domino 5.05) standards support

### Users already in Domino Directory

- Web client authorization

### Maintain one directory for user information

- Easy and well known maintenance
- Phone numbers, e-mail addresses, etc...

### Multiple applications - one directory

- Domino and WebSphere applications accessing the same authentication directory

---

© 2001 IBM Corporation

## Domino LDAP



LDAP code is part of the Domino Server code and is shipped on the Domino Server CD. There is no need to purchase or install it separate from the Domino server. When you create a Domino server, the LDAP code will be automatically loaded at the same time as the rest of the server code is loaded. Even if you do not indicate that you will be using LDAP.

Domino supports LDAP V2 & V3 level of standards. The (Dom 4.6) after V2 and (Dom 5.05) after V3 is to show which version of LDAP applies most to which version of Domino. However, Domino 5.05 does support both V2 & V3 for compatibility purposes. For the purpose of these discussions, the biggest difference between the two LDAP versions is that LDAP V3 is required for Single Sign-on capability and also allows referrals to other LDAP directories.

So if it is part of the Domino Server code how do you get it to run. LDAP services can be started automatically when the Domino server is started or it can be started manually after the Domino server has been started. To have it started automatically, select LDAP in the Internet Directory Services section during the Domino server install. However, in most cases the Domino server will have already been installed you can simply edit the Notes.INI file on the Domino server running the LDAP services and add LDAP to the ServerTasks setting (ServerTasks=Router, Replica, Update, LDAP.....) and re-start the Domino server. If you want to manually start LDAP services on the Domino server you can simply type "load LDAP" on the Domino server's console.

That is the minimum effort it takes to start LDAP services on a Domino server. However to make LDAP services actually useful in your environment, you should read Chapter 14: Setting Up the LDAP Service in the Administering the Domino System manual. Plan to spend some time on this Chapter so that you can understand LDAP and make it function the way you want it to. It is easy to start but it takes some time and effort to make it work efficiently.

Note - LDAP does not by-pass Domino's security for Web clients. Data base security is still controlled by Domino's Access Control List (ACL). If the Web client is to access a data base that requires more than anonymous access, the user must be in the Domino's directory and given ACL authorization to the data base just like a Lotus Notes user.

In an environment where a customer has their users already in the Domino's directory for e-mail purposes it is highly unlikely that they will not want to create another directory with the same data in it. Even if you promise them an easy way to take information out of a Domino directory to input to the new directory. Nor will they be excited about keeping multiple directories current and in-synch. So being able to use LDAP services on an existing Domino directory to authenticate Web clients would be a major advantage for this customer.

---

© 2001 IBM Corporation

## Setting up Domino LDAP Service



### LDAP Defaults



- Port 389
- Name & Password = Yes
- Anonymous = Yes

### LDAP added to ServerTasks line in NOTES.INI file

- Can be loaded manually with Domino console command  
–LOAD LDAP

### Entries in Domino Directory with Internet Password

- Can be accessed by any LDAP client

---

© 2001 IBM Corporation

## Setting up Domino LDAP Service



### Setting up Domino LDAP

In the scenario we are talking about, Domino has been installed and is being used for their e-mail application and the customer does not want to create another directory to authenticate the web client users coming into WebSphere. And the customer wants to authenticate the users via user name and password. We are going to also make the assumptions that all the Domino servers are located in one Domain, there are no non-Domino directories being used to authenticate users, and the web users are already in the Domino directory. For your reference purposes, I have created a bulleted description of the steps and then the following foils show you screen captures of the bulleted information. In this environment, there are no real changes to LDAP defaults to make this work.

We would use the LDAP settings:

Port (389)

Anonymous = Yes

Name & Password = Yes

The NOTES.INI file has been edited so that LDAP was added to the ServerTasks= setting so that it starts automatically when the Domino server starts.

We do have to make sure that we have a user defined in the Domino directory on the server running LDAP that has a internet password. This id will be used during the set-up of WebSphere. In this scenario we created the userid T00 Web. However we could have just as easy used a userid that was already in the Domino directory

---

© 2001 IBM Corporation

## OS/400 LDAP



### **IBM SecureWay Directory for OS/400**

#### **No charge option of OS/400**

- Option 32 of 5769-SS1
- For OS/400 V5R1 included in base operating system

#### **Configured and Managed with Operations Navigator**

#### **Can map users in System Distribution Directory (SDD)**

---

© 2001 IBM Corporation

## OS/400 LDAP



Official name is IBM SecureWay Directory for OS/400, referred to as OS/400 LDAP or Directory Services.

Included free with OS/400 (option number 32 of 5769-SS1). In OS/400 V5R1 or later, Directory Services is part of the base operating system.

All directory server configuration tasks are performed using Operations Navigator.

After OS/400 LDAP is configured and running, you can publish users of the OS/400 System Distribution Directory (SDD) to the LDAP directory.

---

© 2001 IBM Corporation

## Setup WebSphere to use LDAP



### WebSphere Administrative Console

- Console -> Tasks -> Configure Global Security Settings



### Global Security Settings

- General Tab
  - Select Enable Security
- Application Defaults Tab
  - Specify the Domain name for the server in the Realm \*Name field
  - Select Basic (User ID and Password)
- Authentication Mechanism Tab
  - Select Lightweight Third Party Authentication (LTPA)
- User Registry Tab
  - Security Server ID - matches User Name in directory being used
  - Security Server Password - matches User Name's Internet password
  - Directory Type - Domino 5.0 or SecureWay
  - Host - Fully qualified name for server running the LDAP services

© 2001 IBM Corporation

## Setting up WebSphere to use LDAP



Start the WebSphere Administrative Console and select Console - Tasks - Configure Global Security Settings

- General tab
  - Select Enable Security (turn it on)
- Application Defaults tab
  - Type in the internet Domain Name for the Domino server that will be running LDAP services in the Relm \*Name field. Be sure not to include the Domino server name.
  - Select Basic (User ID and Password). By selecting Basic it will prompt the user for their User Name and Password and validate them against a directory entry for the user.
- Authentication Mechanism tab
  - Select Lightweight Third Party Authentication (LTPA). This will identify to WebSphere that it will be using LDAP APIs to access the directory we will be identifying on the next tab.
    - Note: Token Expiration specifies how long the token will be good from the time it is generated - absolute time. 30 minutes may be short.
- User Registry tab
  - Before WebSphere can send an authentication request it must initialize a session with a directory. To do this initialization, it must use a user that has an entry in the directory. If using Domino, this user must have an internet password.

This screens are shown on the following four pages.

© 2001 IBM Corporation

## WebSphere Global Security Settings



**Set Global Security Wizard**

**General**  
Enable WebSphere security (requires restarting the administrative server).

General | Application Defaults | Authentication Mechanism | User Registry

☒ **Enable Security**

Security Cache  
\* Security Cache Timeout:  seconds

< Back   Next >   Finish   Cancel

## WebSphere Global Security Settings



**Set Global Security Wizard**

**Application Defaults**  
Set security defaults for all applications.

General | Application Defaults | Authentication Mechanism | User Registry

Realm  
\* Name:

Challenge Type  
☐ None  
☒ Basic (User ID and Password)  
☐ Certificate  
    ☐ Default To Basic  
☐ Custom  
    \* Login URL:   
    \* Relogin URL:   
☐ Use SSL to connect client and Web server

\* - indicates a required field

< Back   Next >   Finish   Cancel

## WebSphere Global Security Settings



**Set Global Security Wizard**

**Authentication Mechanism**  
Specify how to authenticate clients when they try to access applications.

General | Application Defaults | **Authentication Mechanism** | User Registry

Authentication Mechanism

☐ Local Operating System

☒ Lightweight Third Party Authentication (LTPA)

\* Token Expiration (minutes)

Generate Keys Import From File Export To File

☐ Enable Single Sign On (SSO)

\* Domain

☐ Limit to SSL connections only

< Back Next > Finish Cancel

## WebSphere Global Security Settings



**Set Global Security Wizard**

**User Registry**

General | Application Defaults | Authentication Mechanism | **User Registry**

LDAP

\* Security Server ID

\* Security Server Password

\* Directory Type  Advanced...

\* Host

Port

Base Distinguished Name

Bind Distinguished Name

Bind Password

☐ Use SSL to connect to directory

\* - indicates a required field

< Back Next > Finish Cancel

## Agenda



### Overview

- Collaborative Commerce - Why Domino & WebSphere?
- Points of Integration

### Integration Topics

- HTTP Options
  - ─ Domino HTTP Server Support
    - Configure WebSphere to use Domino HTTP server
  - ─ OS/400 HTTP Server Support
    - Configure Domino to use OS/400 HTTP server
- Authentication & Directory Sharing
- [Single Sign-On](#)



### Summary / Resources

---

© 2001 IBM Corporation

## Agenda - Single Sign On



We now discuss Single Sign On or SSO, specifically in the context of WebSphere and Domino.

What is Single Sign On?

Basically, most people are trying to minimize both the number of userids and passwords that need to be remembered, and the number of times that the user is asked for this information, while maintaining an acceptable level of security. And administrators are trying to minimize the amount of work it takes to coordinate and manage the policy.

---

© 2001 IBM Corporation

## Why use Single Sign On?



### Web browsers authenticate only once

- Even when accessing Domino and WebSphere applications
- Multiple logons not required
- Gain access to all Domino servers and WebSphere Application servers
  - ─ In the same DNS

**Domino and WebSphere Application servers share authentication information.**

---

© 2001 IBM Corporation

## Why use Single Sign On?



SSO support allows Web browsers to authenticate once when accessing both Domino resources, such as documents in a Domino database, and WebSphere Application Server resources, such as HTML, JSPs, servlets and EJBs.

Web browsers can authenticate once to a Domino server or WebSphere application server, then access any other Domino servers or WebSphere application servers in the same DNS domain that are enabled for Single Sign-On (SSO) without signing-on again. This is accomplished by configuring Domino servers and WebSphere application servers to share authentication information between the servers.

---

© 2001 IBM Corporation

## Requirements for Single Sign On



### All servers must be configured for the same DNS domain

- All WebSphere servers must be 3.5.1 or later
- All Domino servers must be 5.05 or later
  - ─ For iSeries, Domino servers must 5.06a or later
- All servers must share the same LTPA keys
- URLs must include DNS domain (no IP addresses or host names)

### An LDAP server is required

- All servers must share the same user registry for authentication
  - ─ All users must be defined in a single LDAP directory
- Either Domino or OS/400 SecureWay LDAP server can be used on iSeries

### Web browsers must be configured to accept cookies

### Server time and time zone must be correct

- SSO token expiration is absolute

### Need to secure Domino Databases & WebSphere Applications

© 2001 IBM Corporation

## Requirements for Single Sign On



All servers must be configured for the same DNS domain because that's the only place cookies are sent. And a fully qualified server name is only way a Web browser knows what cookies to send. For example, if the DNS domain is specified to be "mycompany.com", then SSO will be effective with any Domino or WebSphere application server that serves the "mycompany.com" domain such as "x.mycompany.com" and "y.mycompany.com".

Domino R5.0.6a for iSeries 400 (or later) and Domino R5.0.5 (or later) for other platforms are supported. A Notes client R5.0.5 (or later) is required for configuration of the Domino server for SSO. Authentication can be shared across multiple Domino domains.

WebSphere Application Server V3.5.1 (or later) for all platforms is supported. Authentication can be shared across multiple WebSphere administrative domains. Both the WebSphere Application Server Standard and Advanced Editions are supported. Basic authentication (user ID and password) using either Basic or Custom Challenge Types is supported. Permissions for either all authenticated users or groups of users is supported. If you are using the Domino Directory for authentication, and have not specified a Base Distinguished Name during setup permissions for individual users is also supported.

All servers must share the same user registry, accessible using LDAP. A Domino Directory (configured for LDAP access) or other LDAP directory can be used for the user registry. The LDAP directory product must be supported by WebSphere Application Server. This includes both Domino and all IBM SecureWay LDAP directory servers. All users must be defined in a single LDAP directory. Connecting more than one directory together using LDAP referrals is not supported. Using multiple Domino Directory Assistance documents to access multiple directories is not supported.

The user's Web browser must be configured to accept cookies since the authentication information that is generated by the server is transported to the Web browser in a cookie. The cookie is then used to propagate the user's authentication information to other servers, relieving the user from entering the authentication information for every request to a different server.

Time zones must be coordinated or valid token may be considered expired. And servers must have the set of keys in common.

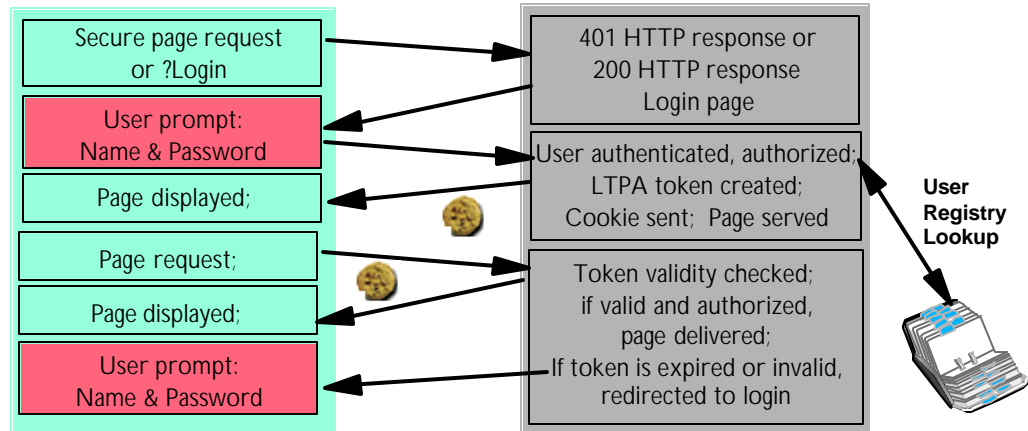
© 2001 IBM Corporation

## How LTPA SSO Authentication Works



### Web Browser

### HTTP Server



© 2001 IBM Corporation

## How LTPA SSO Authentication Works



User requests secure resource. Server (Domino or WebSphere) causes prompt to be issued, asking for name and password. Server authenticated, using common directory, and creates LTPA token, which is sent in a cookie with the page. When user returns to this server or another, a cookie is passed, the server checks it, and knows the user is already authenticated.

Either Domino or WebSphere will generate LTPA token and cookie acceptable to both. User is only prompted once.

© 2001 IBM Corporation

## Configure SSO for WebSphere Application Server



### Before configuring SSO ensure the following as been done:

- Install WebSphere Application Server version 3.5.1 or later
- Configure WebSphere Application servers
- Verify WebSphere Application servers are configured correctly
  - ▀ Access application resource using Web browser
- Verify LDAP directory is available and has at least one user
  - ▀ Using ldapsearch in QShell (strqsh or qsh)
    - ldapsearch -h systemname -p portnumber objectclass=\*

---

© 2001 IBM Corporation

## Configure SSO for WebSphere Application Server



To use SSO with Domino and WebSphere application servers, you must first configure SSO for WebSphere. SSO for WebSphere allows authentication information to be shared across multiple WebSphere administrative domains and with Domino servers.

If you would like to provide SSO to WebSphere application servers in more than one WebSphere administrative domain, you'll need to configure each of the administrative domains to use the same DNS domain, user registry (using LDAP) and a common set of LTPA keys.

Verify that the application servers are configured correctly by trying to access application resources using a Web browser.

Verify the LDAP directory you are going to use is available and configured with at least one user. In the configuration, WebSphere Application Server requires access to the LDAP directory. You can use the Domino Directory or another LDAP directory.

---

© 2001 IBM Corporation

## Configure Single Sign On - WebSphere



### Steps to configure WebSphere for SSO:

- Security Enabled
- Challenge Type set to Name and Password
- Authentication Method to set to LTPA
- Single Sign On enabled & LTPA keys Generated and Exported
- User Registry setup tab
  - Domino or OS/400 SecureWay

---

© 2001 IBM Corporation

## Configure Single Sign On - WebSphere



In order to use Single Sign On between Domino and WebSphere you must enable security on the WebSphere and Domino applications.

To prepare to use the Single Sign-On (SSO) abilities for WebSphere, you must update the WebSphere Global Security configuration with Single Sign-On enabled, and re-generate and export the LTPA keys to be used when you configure Lotus Domino for AS/400 for Single Sign-On.

Note: Before you configure Domino for Single Sign-On with WebSphere, you must configure WebSphere first because the LTPA keys generated by WebSphere have to be imported into Domino.

---

© 2001 IBM Corporation

## Enable SSO, Generate/Export LTPA keys



The screenshot shows the 'Set Global Security Wizard' dialog box with the 'Authentication Mechanism' tab selected. The title bar reads 'Set Global Security Wizard'. Below the title bar, the text 'Authentication Mechanism' is followed by the instruction 'Specify how to authenticate clients when they try to access applications.' The dialog has four tabs: 'General', 'Application Defaults', 'Authentication Mechanism', and 'User Registry'. The 'Authentication Mechanism' tab contains the following options:

- ☐ Local Operating System
- ☒ Lightweight Third Party Authentication (LTPA)
  - \* Token Expiration (minutes): 30
  - Buttons: Generate Keys, Import From File, Export To File
- ☒ Enable Single Sign On (SSO)
  - \* Domain: itsoroch.ibm.com
  - ☐ Limit to SSL connections only

At the bottom of the dialog are four buttons: '< Back', 'Next >', 'Finish', and 'Cancel'.

## Enable SSO, Generate/Export LTPA keys



To enable SSO in WebSphere, you must check the **Enable Single Sign On (SSO)** box to enable SSP and authentication information to be placed in HTTP cookies.

Set Domain to the domain portion of your fully qualified Internet name for the system running your WebSphere administrative domain.

You also need to configure the LTPA keys to be used by the WebSphere administrative domain that you are configuring. To complete the security configuration for SSO, you need to export the LTPA keys to a file. This file is used later when importing keys during the configuration of additional WebSphere administrative domains and when configuring SSO for Domino.

## Configure Single Sign On - Domino



### Steps to configure Domino for SSO:

- Update Domino server document:
  - Enable basic authentication for Web browsers
  - Set Session Authentication to Multi-Server
- Create Web SSO Configuration document
  - Import LTPA keys from WebSphere

---

© 2001 IBM Corporation

## Configure Single Sign On - Domino



Configuring SSO for Domino is accomplished by selecting a new multi-server option in the Domino Server document for session-based authentication. You must also create a new domain-wide configuration document in the Domino Directory called the Web SSO Configuration document. The Web SSO Configuration document, which should be replicated to all Domino servers participating in the SSO domain, is encrypted for participating Domino servers and contains a shared secret used by Domino servers for authenticating user credentials.

---

© 2001 IBM Corporation

## Session Authentication = Multi-Server



|  |          |       |              |   |      |               |                       |
|--|----------|-------|--------------|---|------|---------------|-----------------------|
| Basics   | Security | Ports | Server Tasks | Internet Protocols  | MTAs | Miscellaneous | Transactional Logging |
| <div>HTTP   Domino Web Engine   IIOP   LDAP   NNTP</div>   |          |       |              |   |      |               |                       |
| <b>HTTP Sessions</b><br>Session authentication: <input checked="" type="radio"/> Multi-server <input type="radio"/> Single-server  |          |       |              | <b>Java Servlets</b><br>Java servlet support:<br>Servlet URL path:<br>Class path:<br>Servlet file extensions:<br>Session state tracking:<br>Idle session timeout:<br>Maximum active sessions:<br>Session persistence: |      |               |                       |
| <b>Generating References to this Server</b><br>Does this server use IIS? <input checked="" type="radio"/> No <input type="radio"/> Yes<br>Protocol: <input checked="" type="radio"/> http <input type="radio"/> https<br>Host name: <input type="text"/><br>Port number: <input type="text"/> 8004 |          |       |              | <b>POST Data</b><br>Maximum POST data (in kilobytes):<br>File compression on upload:  |      |               |                       |
| <b>Memory Caches</b><br>Maximum cached commands: <input type="text"/> 128<br>Maximum cached designs: <input type="text"/> 128<br>Maximum cached users: <input type="text"/> 64<br>Cached user expiration interval: <input type="text"/> 120 seconds  |          |       |              |   |      |               |                       |

© 2001 IBM Corporation

## Session Authentication = Multi-Server



## Web SSO Configuration Document



Save and Close Keys...  
Create Domino SSO Key  
Import WebSphere LTPA Keys

Web SSO LtpaToken

Basics Administration

| Token Configuration   |                                  | Token Expiration      |    |
|-----------------------|----------------------------------|-----------------------|----|
| Token Name:           | LtpaToken                        | Expiration (minutes): | 30 |
| Token Domain:         | .ITSOROCH.IBM.COM                |                       |    |
| Participating Servers |                                  |                       |    |
| Domino Server Names:  | Dom\WASxx\Domxx                  |                       |    |
| WebSphere Information |                                  |                       |    |
| LDAP Realm:           | Dom\WASxx.ITSOROCH.IBM.COM\38901 |                       |    |
| LTPA Version:         | 1.0                              |                       |    |

© 2001 IBM Corporation

## Web SSO Configuration Document



Remember that before you can configure Domino for SSO with WebSphere, you need to configure WebSphere first because the LTPA keys generated by WebSphere have to be imported into Domino in the Web SSO Configuration document.

Once you import the keys, the Web SSO document is automatically updated to reflect the information in the LTPA keys file you just imported. Fill in the remaining fields as follows:

- **Token Expiration:** The number of minutes a token can exist before expiring. Note, A token does not expire based on inactivity, it is valid for only the number of minutes specified from the time of issue.
- **Token Domain** - The DNS domain portion of your fully qualified Internet name of your system. Since all servers participating in SSO must be in the same DNS domain, this value must be same as the Domain value specified when configuring WebSphere Application Server. Note: WebSphere Application Server treats the DNS domain as case sensitive, so insure that the DNS domain value is specified exactly the same, including casing, whenever you use the value.
- **Domino Server Names** - The Domino servers that will be participating in SSO. This document will be encrypted for the creator of the document, the members of the Owners and Administrators fields, and the servers specified in this field. Note: You must specify a fully qualified Domino server name here. For example, MyDominoServer/MyOu. The Domino server name that you specify here must also match the name of the Home/mail server in the currently active Location document on your Notes client. (Some documentation mentions the Connection document at this point, which is wrong).
- **LDAP Realm** - The fully qualified TCP/IP host name of the LDAP server. This field is initialized from the information provided in the LTPA keys file. Note: You will only need to change this value if an LDAP server port value was specified for the WebSphere administrative domain. If a port was specified, a \ must be inserted in the value before the colon. For example, replace mymachine.mydomain.ibm.com:389 with mymachine.mydomain.ibm.com\389.

© 2001 IBM Corporation

## Programmatic Single Sign On



### **New SessionToken property in `lotus.domino.Session`**

- `getSessionToken` allows Java access to LTPA token

### **Extensions to `NotesFactory` class**

- `Token` and `omg.org.SecurityLevel2.Credentials` parameters added to `createSession`
- Can be used while creating Notes session from WebSphere

### **Documented in 5.0.5 Release Notes**

---

© 2001 IBM Corporation

## Programmatic Single Sign On



So far, we have only talked about single sign on from the web browser perspective. But there are also ways to take advantage of it in the back end. These new properties allow access and use of the existing LTPA token to establish an authenticated session with Domino from WebSphere.

---

© 2001 IBM Corporation

## Agenda



### Overview

- Collaborative Commerce - Why Domino & WebSphere?
- Points of Integration

### Integration Topics

- HTTP Options
  - Domino HTTP Server Support
    - Configure WebSphere to use Domino HTTP server
  - OS/400 HTTP Server Support
    - Configure Domino to use OS/400 HTTP server
- Authentication & Directory Sharing
- Single Sign-On



### Summary / Resources

© 2001 IBM Corporation

## Additional Information - Web Resource



### IBM Domino for iSeries:

- Updated information on Domino for iSeries
- <http://www.iseries.ibm.com/Domino>

### IBM WebSphere for iSeries:

- Updated information on Domino for iSeries
- <http://www.iseries.ibm.com/WebSphere>

### Domino & WebSphere for iSeries Integration:

- <http://www-1.ibm.com/servers/eserver/series/domino/domwas.htm>

### Lotus WebSphere Integration:

- Lotus information on WebSphere Integration
- <http://www.lotus.com/WebSphere>

### IBM ITSO:

- Redbooks, redpapers, ITSO presentations
- <http://www.redbooks.ibm.com/>

### Technical Studio:

- Watch for new topic "Configuring the OS/400 LDAP Server"
- <http://www.iseries.ibm.com/TStudio>

© 2001 IBM Corporation

## Additional Information - Redbooks



Domino and WebSphere Together, SG24-5955

- currently being updated at Lotus ITSO center

Domino and WebSphere Integration on iSeries, SG24-6223

- projected availability, 3Q20001
- second stage of residency scheduled July 9 - Aug 10 in Rochester

WebSphere V3.5 Handbook, SG24-6161



# Redbooks

International Technical Support Organization

© 2001 IBM Corporation